

Introduction to Vector Field Visualization



David Kao

Han-Wei Shen

NASA Advanced
Supercomputing (NAS) Division
NASA Ames Research Center

Computer Science and
Engineering Department
The Ohio State University

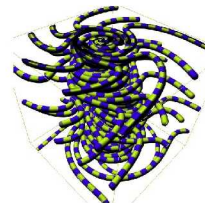
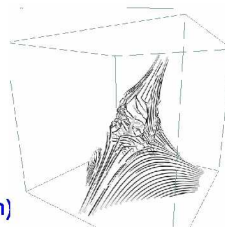
*A Tutorial for
IEEE Pacific Visualization Symposium 2010*



Topics Today

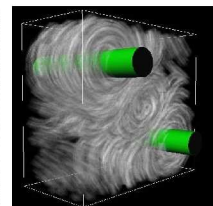
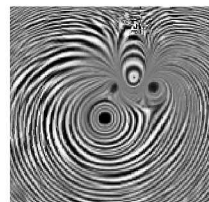
▪ Vector Field Visualization (Kao)

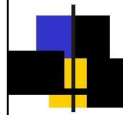
- CFD/numerical flow visualization
- Particle tracking algorithms
- Instantaneous vs time-dependent visualization
- Unsteady flow volumes



▪ Surface Flow Texture Visualization (Shen)

- Steady and time-dependent flow textures
 - LIC, UFLIC, LEA, IBFV, and IBFVS
- 2D and 3D seed placement algorithms
 - Energy-based methods
 - Evenly-spaced methods
 - Topology-based methods





Introduction to Vector Field Visualization

David Kao

NASA Advanced Supercomputing (NAS) Division
NASA Ames Research Center

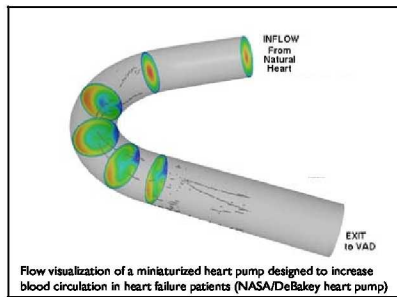


Vector Field Visualization - Applications

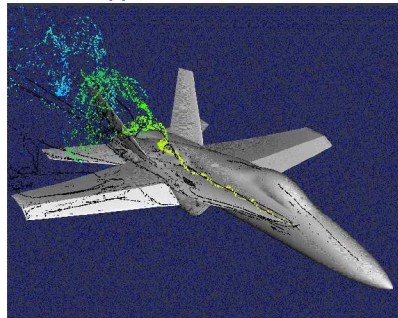
Vector fields are found in many real-world applications:

- Study of flow around an aircraft
- Blood flow in our heart chambers
- Ocean circulation models
- Severe weather predictions

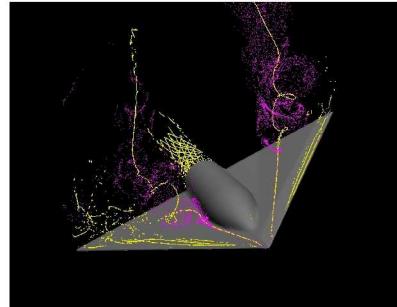
The vector fields from these various applications can be visually depicted using a number of techniques such as particle traces and advecting textures



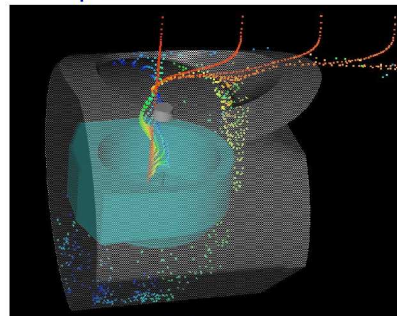
Medical Applications



Defense

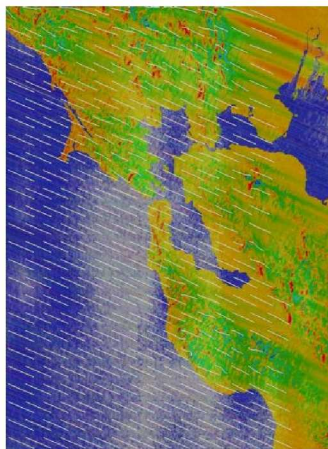


Aerospace

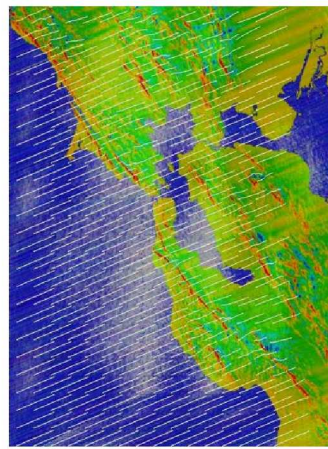


Astronomy (SOFIA telescope)

Summer Condition



Winter Condition



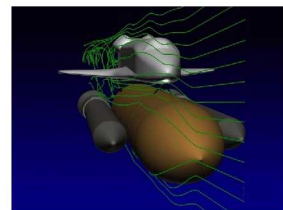
- Simulated airflow over across a mountainous portion of Northern California under two contrasting meteorological conditions
- Visualization of cloud texture overlay with streamlines
- Streamline paths are computed based on the input wind field
- 4th-order Runge-Kutta integration scheme is used to advect the particles
- Animations depicting flow over this geographical location provide immediate assistance in decision support and crisis management

Vector Field Visualization - Problem

- A vector field $\mathbf{V}(\mathbf{p})$ is given for discrete points \mathbf{p} where \mathbf{p} lie in either a 2D or 3D grid
- 2D vector field visualization is straightforward
- 3D vector field visualization is challenging due to 3D perspective
- Time-dependent flow visualization has additional challenges
- A vector field $\mathbf{V}(\mathbf{p}, t)$ is given for discrete points \mathbf{p} and at many time steps
- Time-dependent flow visualization is also known as unsteady flow visualization

Visualization Techniques

- **Geometry-based methods**
rendering primitives built from particle trajectories
 - 1D: streamlines, pathlines, streaklines
 - Line variations: tubes, ribbons
 - 2D: stream surfaces
 - 3D: flow volumes

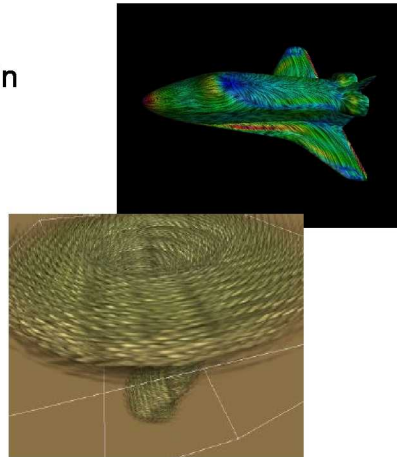


Visualization Techniques - II

■ Texture-based methods

shading every pixels/voxels in the visualization

- LIC
- Texture Splats
- Chameleon

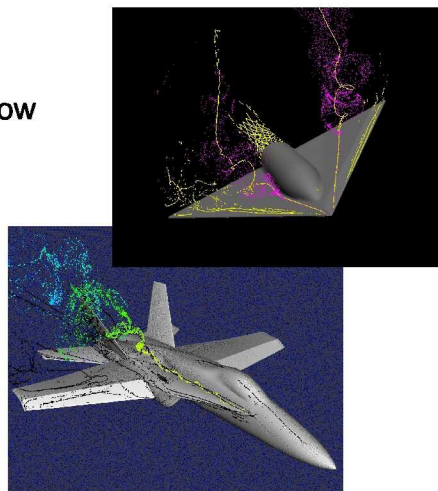


Visualization Techniques - III

■ Topological-based methods

extracting features based on flow topology

- Critical points
- Vortex cores
- Skin-friction lines



Computational Fluid Dynamics

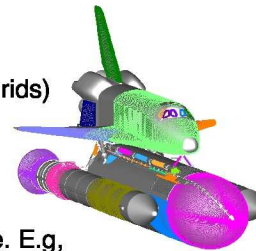
- A computational technology that enables one to study the dynamics of things that flow
- Runs on computer systems to model fluid flows using mathematical modeling, numerical methods, and software tools for pre-processing and post processing
- The wind tunnel has been traditionally used to simulate physical flows
- With the advent of more powerful computers, CFD is now the preferred means of flow simulation before final experimental testing (if any)



A Space Shuttle orbiter model in NASA Ames Research Center's 40x80 ft wind tunnel.
NASA Ames Image Library (Photo: A. Melliar, 1975)

Typical Steps in a CFD Analysis: Pre-Processing

- Geometry processing (CAD surface geometry, surface domain decomposition)
- Grid generation (surface grids, hole cutting, volume grids)
- Flow solver input preparation (solver input files)
- Boundary conditions (per grid basis, at each grid face. E.g, periodic, in-flow, out-flow, no-slip/viscous)
- Flow conditions (e.g., Mach number, Reynolds number, angle of attack)
- Numerical methods (parameters for the Navier-Stokes Eqs)



Typical Steps in a CFD Analysis: **Flow Computation**

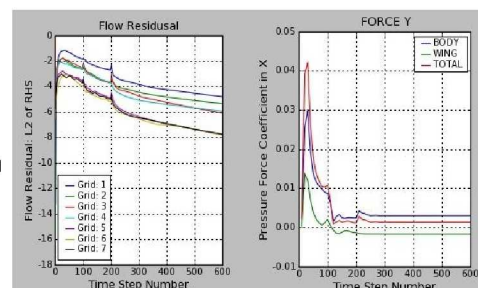
- Selection of physics model
 - inviscid/viscous/turbulence
 - body dynamics
 - no. of species
- Selection of numerical methods parameters
 - accuracy
 - stability
 - convergence
- Run the simulation



A corner view of the 51,200 core SGI Altix ICE system housed at the NASA Advanced Supercomputing facility.
 Photo Credit: NASA Ames Research Center/Marco Libero

Typical Steps in a CFD Analysis: **Post-Processing**

- Convergence analysis
 - Flow residual history plots
 - Turbulence residuals (if any)
- Forces and moments computation
 - Coefficient plots
- **Flow visualization**



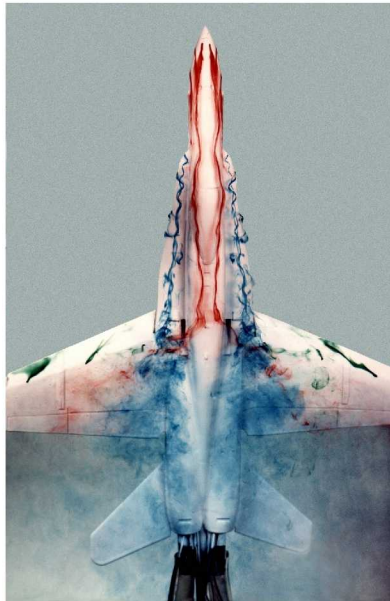
As computation power continues to increase over the past decade, flow simulations require less amount of time to run.

Nowadays, the pre-processing and post-processing stages consume most of the analysis time.

Common Methods in Experimental Flow Visualization - Adding Foreign Material

- **Streaklines:** dye injected from a fixed position. By injecting the dye for a period of time, a line of dye in the fluid is visible
- **Timelines:** a row of small particles (hydrogen bubbles) released at right angles to flow. The motion of the particles shows the fluid behavior
- **Pathlines:** small particles (magnesium powder in liquid; oil drops in gas) are added to the fluid. Velocity is measured by photographing the motion of the particles with a known exposure time

Experimental Flow Visualization



NASA Dryden Flight Research Center Photo Collection
<http://www.dfrc.nasa.gov/gallery/photos/index.html>
 NASA Photo ECRN-33286-03 Date: 1985

1/48-scale model of an F-18 aircraft in Flow Visualization Facility (FVF)



Dryden Flight Research Center ECRN-0096-240 Photographed 1989
 F-18 HARV smoke and tuft flow visualization.
 Angle of Attack = 30 deg. NASA photo.

NASA Photos (in Public Domain)

Experimental Flow Visualization



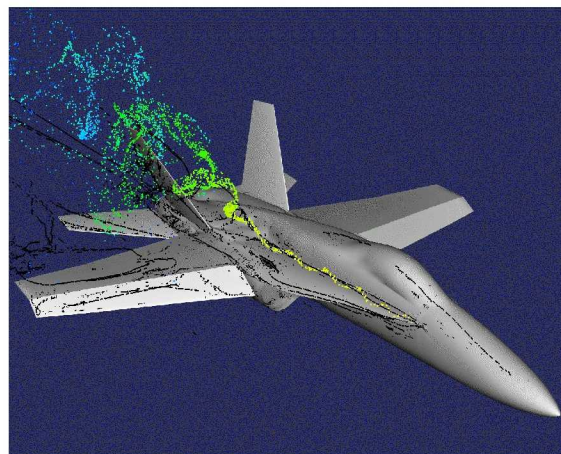
Smoke and laser lighting sheet
(NASA Langley, FS-2001-04-64-LaRC)



Oil flow visualization
(NASA Dryden, IS-97/08-DFRC-02)

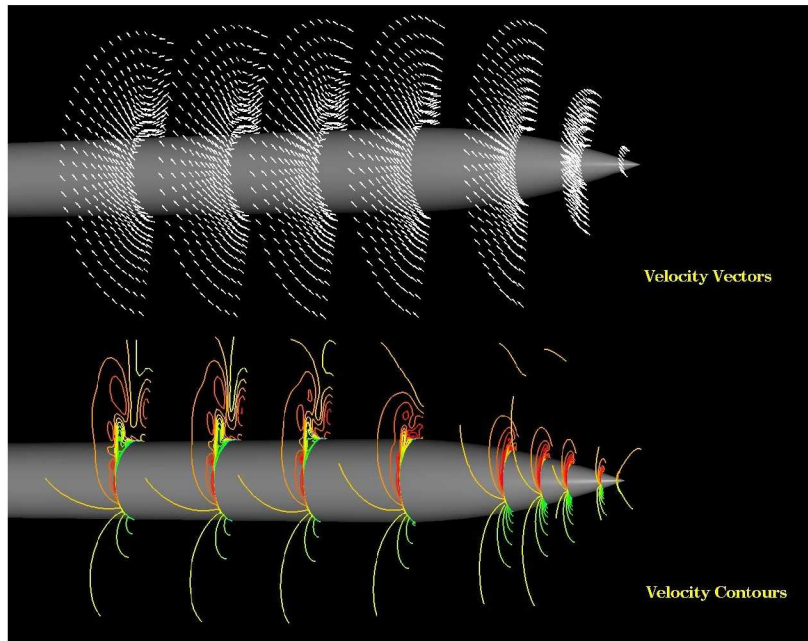
NASA Photos (in Public Domain)

Numerical Flow Visualization

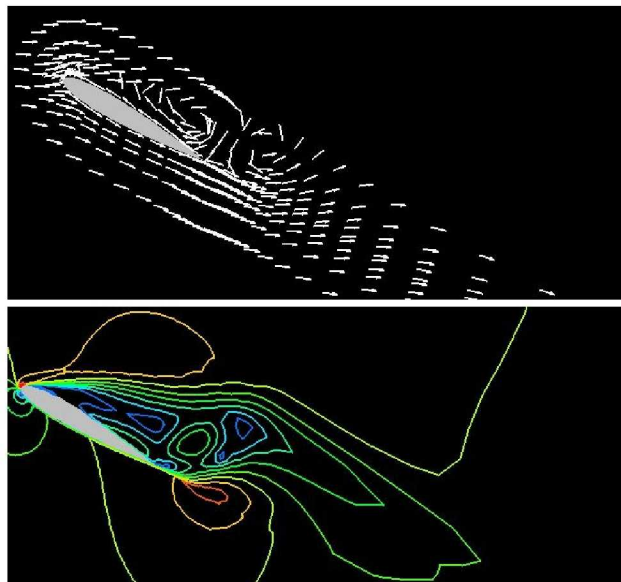


Though numerical flow visualization is not able to totally replicate the results from experimental flow visualization, it has been widely accepted as an effective mean to obtain accurate representation of the CFD flow solutions.

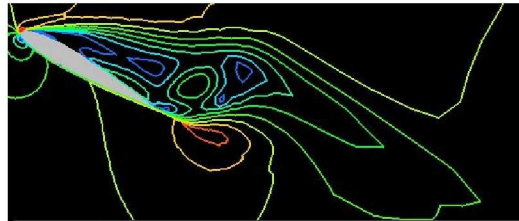
Numerical Flow Visualization – Basic Methods



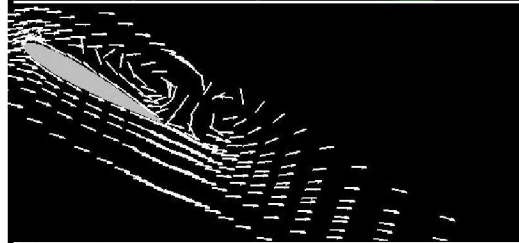
Numerical Flow Visualization – Basic Methods



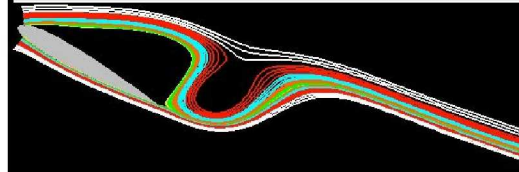
Numerical Flow Visualization – Basic Methods



Velocity Magnitude Contours



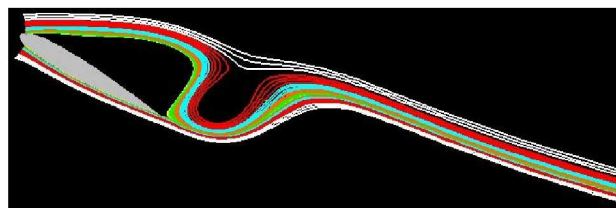
Vector Arrow Plots



Particle Traces - Streamlines

Streamlines

- *Streamline*: a line that is tangential to the instantaneous velocity direction (velocity is a vector, and it has a magnitude and a direction)
- Release a particle into the flow and perform numerical integration to compute the path of the particle

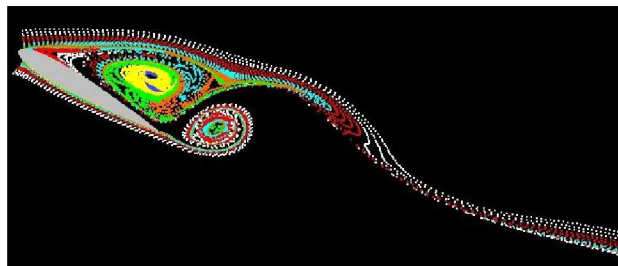


Pathlines

- A *pathline* shows the trajectory of a single particle released from a fixed location (seed point)
- Experimental method: marking a fluid particle and taking a time exposure photo of its motion will generate a pathline
- This is similar to what you see when you take a long-exposure photograph of car lights on a freeway at night

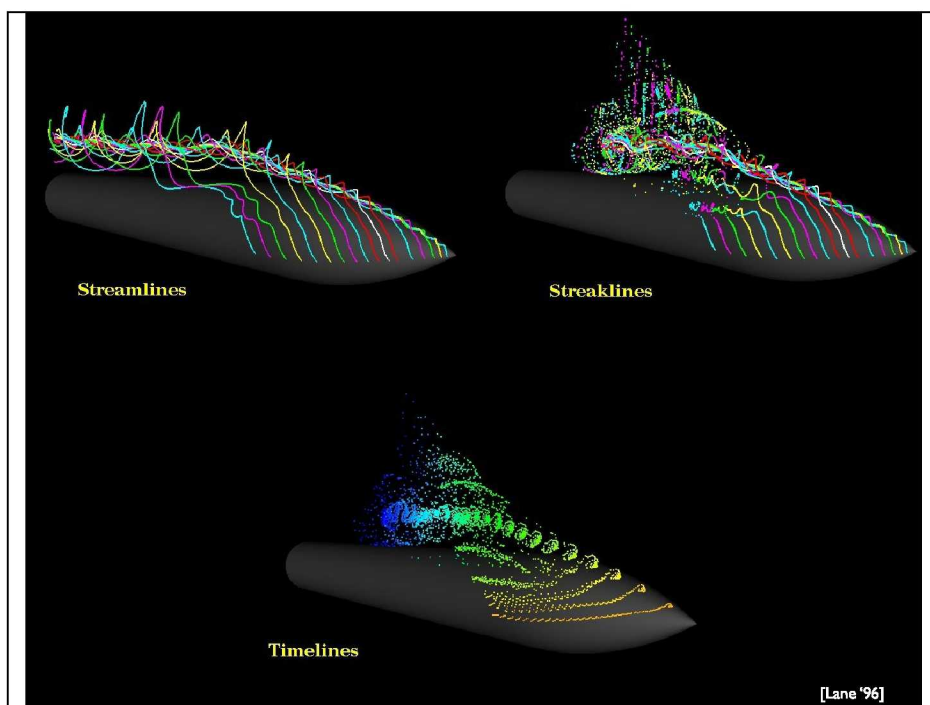
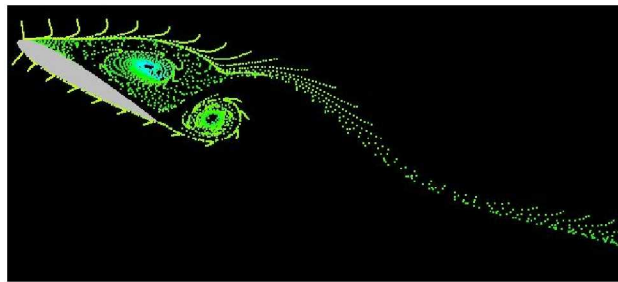
Streaklines

- *Streakline*: a line joining the positions, at an instant in time, of all particles that were previously released from a fixed location (seed point)
- Continuously inject particles into the flow at each time step and track the paths of the particles
- In a steady flow field, streamlines, pathlines, and streaklines are identical. However, they can be very different in unsteady flows



Timelines

- *Timeline*: a line connecting a row of particles that released simultaneously
- Timelines are generated by injecting rows of particles at some fixed time interval



Stream/Path/Streak/Time Lines - Summary

- *Streamline*: a field line tangent to the velocity field at an instant in time
- *Pathline*: shows the trajectory of a single *weightless* particle released from a seed point
- *Streakline*: a line joining the positions, at an instant in time, of all particles released from a seed point
- *Timeline*: a line connecting a row of particles that released simultaneously

In a steady flow field, streamlines, pathlines, and streaklines are identical

Particle Tracing

The path of a massless particle at position \mathbf{p} at time t is determined by:

$$\begin{aligned} \frac{d\mathbf{p}}{dt} &= \mathbf{v}(\mathbf{p}(t)) && \text{for steady flow} \\ \text{or} \\ \frac{d\mathbf{p}}{dt} &= \mathbf{v}(\mathbf{p}(t), t) && \text{for unsteady flow} \end{aligned}$$

The actual particle position at any given time can be determined by integrating the above equation (assuming that $\mathbf{p}(0)$ is given):

$$\mathbf{p}(t + \Delta t) = \mathbf{p}(t) + \int_t^{t+\Delta t} \mathbf{v}(\mathbf{p}(t), t) dt$$

Numerical Integration Methods

- Euler's Method: (Not recommended due to inaccuracy)

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{v}(\mathbf{p}_k) \Delta t \quad \mathbf{p}_0 = \text{seed point}$$

- 2nd Order Runge-Kutta Method: (commonly used)

$$\mathbf{p}^* = \mathbf{p}_k + \mathbf{v}(\mathbf{p}_k) \Delta t$$

$$\mathbf{p}_{k+1} = \mathbf{p}_k + [\mathbf{v}(\mathbf{p}_k) + \mathbf{v}(\mathbf{p}^*)] \Delta t / 2$$

- Higher Order Methods: (4th order RK and others) (recommended)

$$\mathbf{a} = 2\Delta t \mathbf{v}(\mathbf{p}_k), \quad \mathbf{b} = 2\Delta t \mathbf{v}(\mathbf{p}_k + \mathbf{a}/2), \quad \mathbf{c} = 2\Delta t \mathbf{v}(\mathbf{p}_k + \mathbf{b}/2)$$

$$\mathbf{d} = 2\Delta t \mathbf{v}(\mathbf{p}_k + \mathbf{c}/2), \quad \mathbf{p}_{k+1} = \mathbf{p}_k + (\mathbf{a} + 2\mathbf{b} + 2\mathbf{c} + \mathbf{d})/6$$

A good comparison of several integration methods can be found in [Teitzel et al. '97]

Notes on Particle Integration

- The accuracy of particle tracing depends highly on the integration method used.
- Flow solvers are often second-order accurate in time. Particle integration methods should be at least third-order or higher.
- Though multi-stage integration methods such as 4th RK are commonly used, they require velocity data to be interpolated between the two consecutive time steps.
- [Darmofal & Haimes '96] recommended using a fourth-order interpolant instead of linear interpolation:

$$u^{n+1/2}(x) = \frac{5}{16} u^{n+1}(x) + \frac{15}{16} u^n(x) - \frac{5}{16} u^{n-1}(x) + \frac{1}{16} u^{n-2}(x)$$

Basic Particle Tracing Algorithm

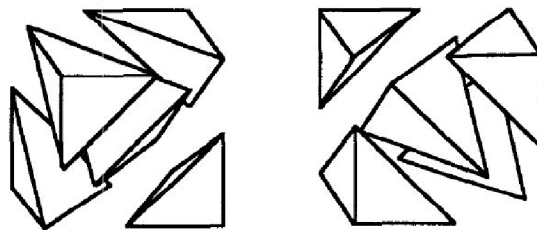
1. *Specify the seed point $\mathbf{p}(0)$, $t=0$*
2. *Perform cell search to locate the cell that contains $\mathbf{p}(t)$*
3. *Interpolate the velocity field to determine the velocity at $\mathbf{p}(t)$*
4. *Advance the particle at $\mathbf{p}(t)$ using either RK4 or a higher-order method*
Note: cell search and velocity interpolation are needed for the intermediate points \mathbf{p}^ during the integration*
5. *Repeat from Step 2 until the particle leaves the flow field*

Computational VS Physical Space

- Particle Tracing in Computational Space
 - Map the given curvilinear grid and the associated velocity field onto a uniform grid by computing and then transforming Jacobian matrices
 - Since a typical grid is large, transformation is performed on-the-fly
 - Main advantage: cell search is eliminated
 - Disadvantages: Particle traces may not be accurate because the transforming Jacobian matrices are only approximations
- Particle Tracing in Physical Space
 - More accurate because the interpolation and integration steps are performed in Cartesian space
 - Velocity field transformation is not necessary
 - Cell search is more time consuming, especial in multi-block grids

Cell Search Based on Tetrahedral Decomposition

- Decompose the current hexahedral cell into five tetrahedra
- Odd or even configuration based on the sum of the node's indices (i,j,k)
- Coordinate and velocity interpolation are based on natural coordinates
- Have been shown to gain a speed up factor of up to six times faster!



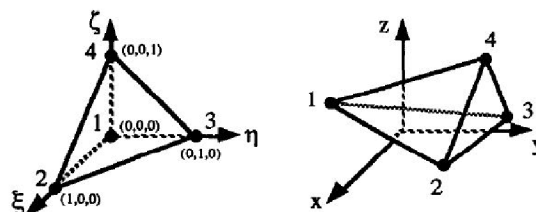
Alternate between two configurations to ensure continuity between cells

[Kenwright & Lane '96]

Cell Search Based on Tetrahedral Decomposition

- A point p is inside the tetrahedron if its natural coordinates satisfy the following conditions:

$$\xi \geq 0, \quad \eta \geq 0, \quad \zeta \geq 0, \quad 1 - \xi - \eta - \zeta \geq 0$$



Tetrahedron geometry in natural (non-dimensional) and physical coordinate spaces

Particle Tracing Issues

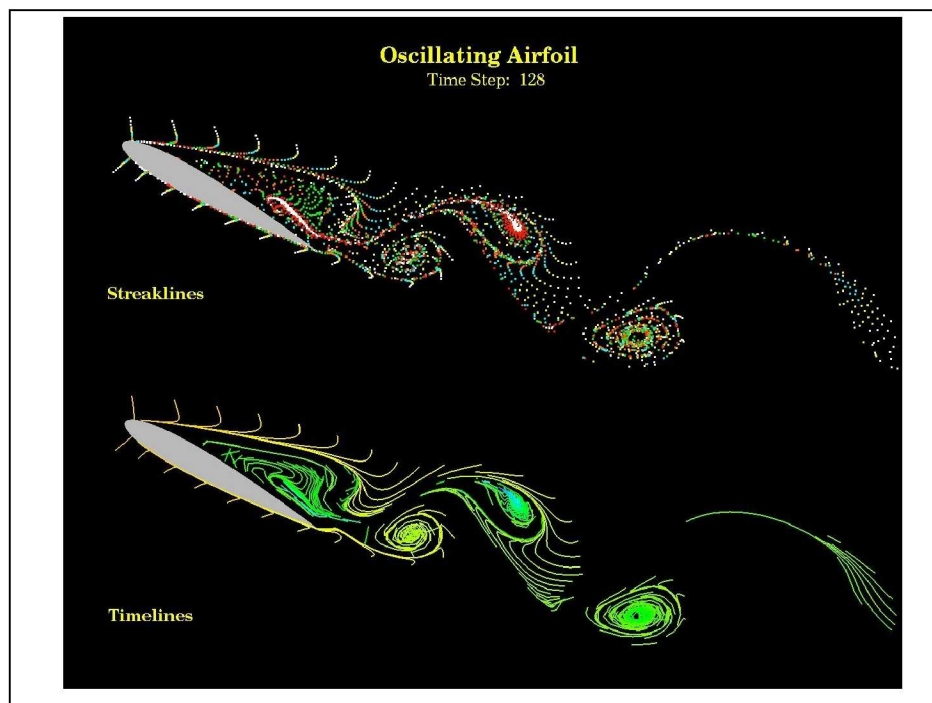
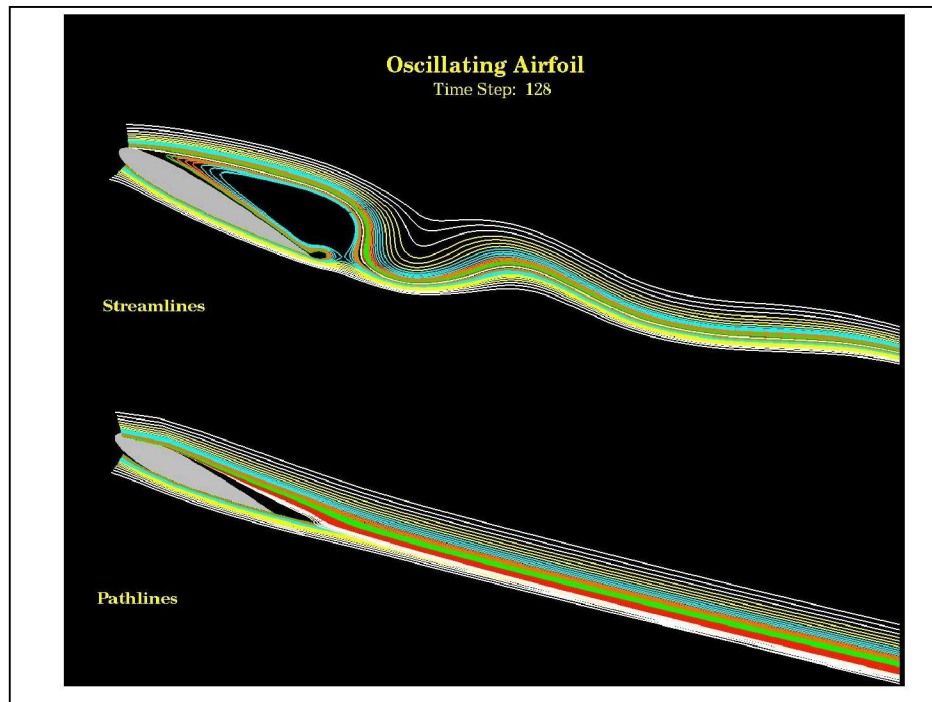
- Adaptive integration step size
 - Variable integration step size, dt , is based on the local velocity vector
- Particle tracing in multi-block grids
 - Overlapping multi-block grids are common in complex grid geometries
 - Particle paths are likely to cross multiple grids (grid jumping is necessary)
- Particle tracking in moving grids
 - Particle position is based on the current grid block and time step
 - The Jacobian matrix needs to consider the grid velocity

$$J = \begin{bmatrix} x_{\xi} & x_{\eta} & x_{\zeta} & x_{\tau} \\ y_{\xi} & y_{\eta} & y_{\zeta} & y_{\tau} \\ z_{\xi} & z_{\eta} & z_{\zeta} & z_{\tau} \\ 0 & 0 & 0 & t_{i+1} - t_i \end{bmatrix} \quad (x_{\tau}, y_{\tau}, z_{\tau}) = \mathbf{p}^{t_{i+1}}(\xi, \eta, \zeta) - \mathbf{p}^{t_i}(\xi, \eta, \zeta)$$

See [Lane '93] for more discussions

Particle Traces - A Comparison

- 2D oscillating airfoil
- Unsteady flow
- Moving grid
- 200 Time Steps
- Data set: Sungho Ko ('95)

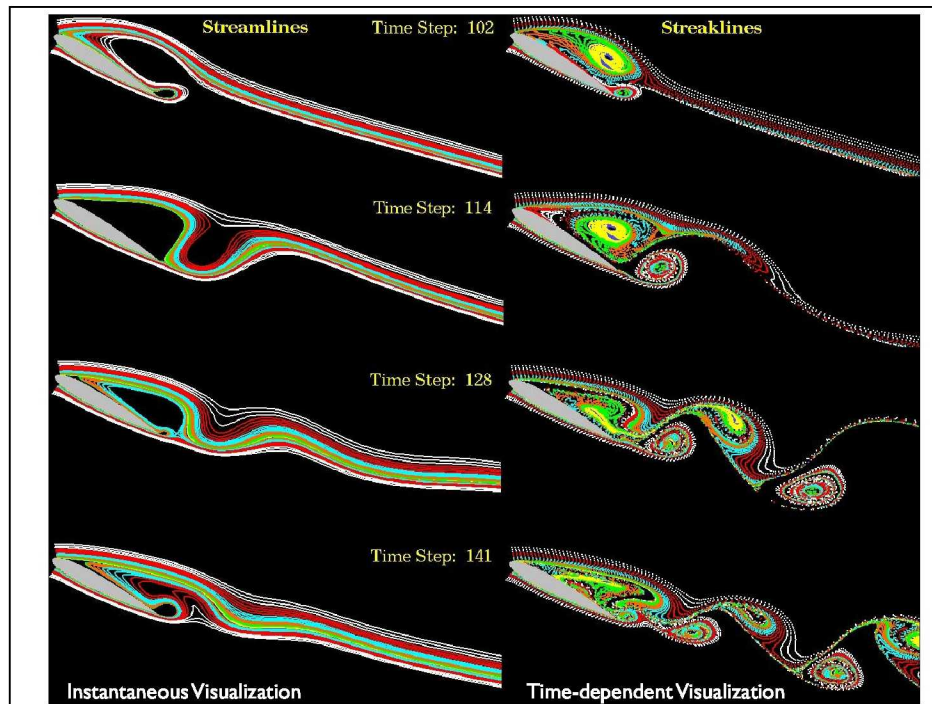


Instantaneous Flow Visualization

- Streamlines, vector plots, contours, and cutting planes
- Calculation is based on one time step of the flow
- Effective for depicting the flow at an instant in time
- Interactive visualization is possible
- Time-variable is not used in the calculation

Time-Dependent Flow Visualization

- Calculation is based on many time steps
- Time variable is used in the calculation
- Effective for depicting time-varying phenomena such as vortex shedding, vortex breakdown, and shock waves
- Require large disk storage
- Interactive visualization is limited



Animation

- Animation is an effective method to depict time-varying phenomena in unsteady flows
- Sometimes, instantaneous visualization techniques are applied to unsteady flow data one time step at a time, then the results are animated
- Instantaneous and time-dependent flow visualization can reveal very different information
- Time-dependent flow visualization should be used to complement instantaneous visualization

Some Extensions of 3D Particle Tracking

Stream surface [Hultquist '92]

Stream ribbon and tube [Ueng, Sikorski and Ma '96]

Stream ball and streak ball [Brill et al. '94]

Dash tube [Fuhmann and Gröller '98]

Time surface [Westermann, Johnson and Ertl '01]

Smoke surface [von Funck et al. '08]

Integral surface [Garth et al. '08]

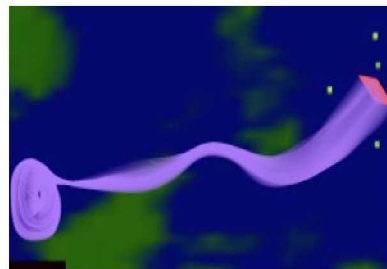
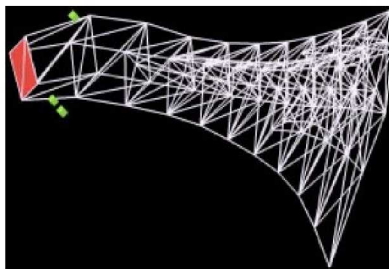
Streak surface [Bürger et al. '09] and [Krishnan et al. '09]

Flow volume [Max, Becker and Crawfis '93]

Unsteady flow volume [Becker, Lane, and Max '95]

Flow Volume

- Advect multiple streamlines from a polygon
- Flow volume is the volume bounded by the 3D streamlines
- Sorting is avoided by assuming monochrome flow volume
- When an edge is too long, it is subdivided at midpoint
- The flow volume is decomposed into tetrahedron cells

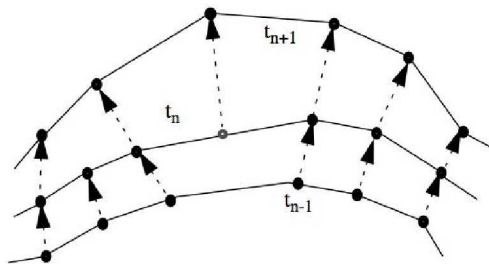


[Max, Becker and Crawfis '93]

Unsteady Flow Volume

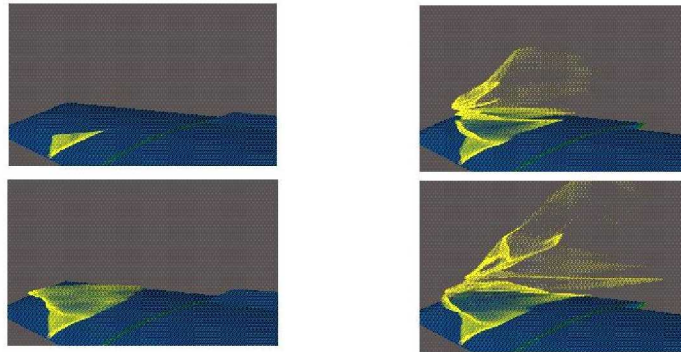
- Advect multiple streaklines from a polygon
- Flow volume is the volume bounded by the 3D streaklines
- Flow volume evolves in time by moving “sideways” as the streaklines shift with the time-varying flow
- All vertices in the flow volume need to be advected over time
- Unsteady flow volume changes more rapidly than steady flow volume

Adaptive Subdivision



- Unlike steady flow volume, where only the last layer of particles (at the flow front) are advected in pseudo time
- Unsteady flow volume requires the ENTIRE flow volume to be reconstructed at each time step
- A new particle is inserted at the midpoint of a segment when a segment becomes too long
- An existing particle is deleted when two adjacent segments become too short

Unsteady Flow Volume Over A Wing With Oscillating Flap



[Becker, Lane and Max '95]

Unsteady Flow Visualization/Large-Scale Data Visualization

- Unsteady flow visualization often has the challenges associated with large-scale data visualization due to the time-varying nature of the data
- Some previous workshops/symposiums on large-scale data visualization:
 - o Symposium on Visualizing Time Varying Data (ICASE/NASA LaRC '95)
 - o Visualizing Large-scale Data, BOF (Visualization '97)
 - o Visualizing Large Datasets: Challenges and Opportunities (SIGGRAPH '99)
 - o Time-Varying Data Visualization Workshop (Supercomputing '05)
 - o Ultra-Scale Visualization Workshop (SC '06 – SC '09)

Large-scale Data Visualization Challenges

- “Time-varying datasets present difficult problems for both analysis and visualization. For example, the data may be **terabytes** in size, distributed across mass storage systems at several sites, with time scales ranging from femtoseconds to eons.”

❖ *David Banks et al., Symposium on Visualizing Time-Varying Data at ICASE/NASA LaRC '95*

- “The output from leading-edge scientific simulations is so voluminous and complex that advanced visualization techniques are necessary to interpret the calculated results. Even though visualization technology has progressed significantly in recent years, we are barely capable of exploiting **terascale** data to its full extent, and **petascale** datasets are on the horizon.”

❖ *Kwan-Liu Ma, Ultra-Scale Visualization at SC'07*

Progress in Large-Scale Data Visualization

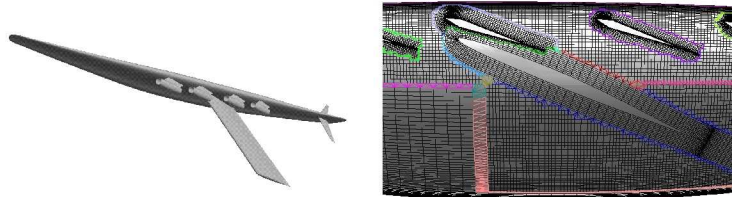
Though, there has been many good progresses made in:

- Flow visualization techniques
- Hardware-assisted/GPU-based visualization
- Out-of-core algorithms
- Parallel algorithms
- Multiresolution and data compression techniques
- High performance computing
- In situ visualization and data reduction

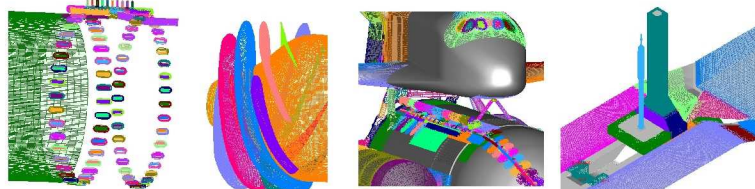
The size of the flow simulation continues to increase ...

Grid Systems of Past and Present

Mid 1990's: 5 - 10 grids, 5 - 10 million volume grid points



Today: 100+ grids, 100+ million volume grid points



Courtesy of William Chan (NASA)

References

- B. Becker, D. Lane, and N. Max. Unsteady flow volumes, *Visualization* 1995.
- M. Brill, H. Hagen, H.-C. Rodrian, W. Djatschin, and S. Klimenko. Streamball techniques for flow visualization, *Visualization* 1994.
- P. Buning. Numerical algorithms in CFD post-processing, In Computer Graphics and Flow Visualization in CFD. VKI Lecture Series, 1989.
- K. Bürger, F. Ferstl, H. Theisel, and R. Westermann. Interactive streak surface visualization on the GPU, *IEEE TVCG*, Vol. 15 (6), November/December 2009.
- D. Darmofal and R. Haines. An analysis of 3-D Particle Path Integration Algorithms, *Journal of Computational Physics*, Vol. 123, 1996.
- A. Fuhrmann and E. Gröller. Real-time techniques for 3D flow visualization, *Visualization* 1998.
- C. Garth, H. Krishnan, X. Tricoche, T. Bobach, and K. Joy. Generation of accurate integral surfaces in time-dependent vector fields, *IEEE TVCG*, Vol. 14 (6), November/December 2008.
- J. Hultquist. Constructing stream surfaces in steady 3D vector fields, *Visualization* 1992.
- D. Kenwright and D. Lane. Interactive time-dependent particle tracing using tetrahedral decomposition, *IEEE TVCG*, Vol 2 (2), June 1996.
- H. Krishnan, C. Garth, and K. Joy. Time and streak surface for flow visualization in large time-varying data sets, *IEEE TVCG*, Vol. 15 (6), November/December 2009.
- D. Lane. "Scientific Visualization of Large-Scale Unsteady Fluid Flows" in Focus on Scientific Visualization, H. Hagen, H. Mueller, G. Nielson, eds., Springer, 1993.
- D. Lane. Visualizing time-varying phenomena in numerical simulations of unsteady flows, *AIAA-96-0048*, 1996.

References

- N. Max, B. Becker, R. Crawfis. Flow volumes for interactive vector field visualization, *Visualization 1993*.
- W. Merzkirch. Flow Visualization. Academic Press, 2nd edition, 1987.
- F. Post and T. Walsum. "Fluid Flow Visualization" in Focus on Scientific Visualization, H. Hagen, H. Mueller, G. Nielson, eds., Springer, 1993.
- C. Teitzel, R. Grosso, and T. Ertl. "Efficient and reliable integration methods for particle tracing in unsteady flows on discrete meshes" in Visualization in Scientific Computing '97, Eurographics.
- S. Ueng, C. Sikorski, and K.-L. Ma. Efficient streamline, streamribbon, and streamtube constructions on unstructured grids, *IEEE TVCG*, Vol. 2 (2), September/October 1996.
- M. van Dyke. An Album of Fluid Motion. The Parabolic Press, 1982.
- W. von Funck, T. Weinkauff, H. Theisel, and H.-P. Seidel. Smoke surfaces: an interactive flow visualization technique inspired by real-world flow experiments, *IEEE TVCG*, Vol. 14 (6), 2008.
- T. Weinkauff, J. Sahner, H. Theisel, and H.-C. Hege. Cores of swirling particle motion in unsteady flows, *Visualization 2007*.
- R. Westermann, C. Johnson, and T. Ertl. Topology-preserving smoothing of vector fields, *IEEE TVCG*, Vol. 7 (3), 2001.
- A. Wiebel and G. Scheuermann. Eyelet particle tracing - steady visualization of unsteady flow, *Visualization 2005*.
- A. Wiebel, X. Tricoche, H. Janicke, and G. Scheuermann. Generalized streak lines: analysis and visualization of boundary induced vortices, *IEEE TVCG*, Vol. 13 (6), Nov/Dec 2007.
- W. Yang. Handbook of Flow Visualization, Hemisphere Publishing, 1989.
- UFAT – www.nasa.nasa.gov/Software/UFAT